Camera Pose Estimation and RANSAC

Srikumar Ramalingam

School of Computing
University of Utah
Presentation Outline

1. Review
2. Pose Estimation
3. RANSAC
Camera Pose Estimation and RANSAC

Srikumar Ramalingam

Review
Pose Estimation
RANSAC

Camera Models and Projection (Reminder)

Let the optical center $O$ be the origin of the camera.

Let $(X^m, Y^m, Z^m)$ be the coordinates of a 3D point $Q$, relative to the world system.

Let the 2D pixel be denoted by $q(u, v, 1)^T$. 
Camera Models and Projection (Reminder)

- Projection of 3D point on the image:

\[
\begin{pmatrix}
u \\
v \\
1
\end{pmatrix} \sim \begin{pmatrix}
K & 0 \\
0^T & -Rt
\end{pmatrix}
\begin{pmatrix}
X^m \\
Y^m \\
Z^m \\
1
\end{pmatrix}
\]

- The following 3 × 3 matrix is the camera matrix:

\[
K = \begin{pmatrix}
k_u f & 0 & k_u x_0 \\
0 & k_v f & k_v y_0 \\
0 & 0 & 1
\end{pmatrix}
\]
The projection matrix that maps 3D points to 2D image is given by:

\[
P = \begin{pmatrix}
K & 0 \\
\end{pmatrix}
\begin{pmatrix}
R & -Rt \\
0^T & 1 \\
\end{pmatrix}
\]

\[
P = \begin{pmatrix}
KR & -KRt \\
\end{pmatrix}
\]

\[
P = KR \begin{pmatrix}
I & -t \\
\end{pmatrix}
\]

Different forms of \underline{projection matrices}
What is Camera Calibration?

- The task refers to the problem of computing the camera matrix $K$.
- In other words, we compute the focal length, principal point, and aspect ratio in the camera matrix.
Forward Projection

\[
\begin{pmatrix}
\ u \\
\ v \\
\ 1
\end{pmatrix}
\sim KR \begin{pmatrix}
\ 1 & -t \\
\end{pmatrix}
\begin{pmatrix}
\ X^m \\
\ Y^m \\
\ Z^m \\
\ 1
\end{pmatrix}
\]
Backward Projection

\[
Q \sim K^{-1}q
\]

\[
Q \sim K^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}
\]

Inverse is also 3x3 homogeneous coordinates
Presentation Outline

1. Review
2. Pose Estimation
3. RANSAC
What is pose estimation?

The problem of determining the position and orientation of the camera relative to the object (or vice-versa).

Left: Camera Image, Right: 3D model of the world
What is pose estimation?

The problem of determining the position and orientation of the camera relative to the object (or vice-versa).

We use the correspondences between 2D image pixels (and thus camera rays) and 3D object points (from the world) to compute the pose.
We consider that the camera is calibrated, i.e. we know its calibration matrix $K$.

We are given three 2D image to 3D object correspondences. Let the 3 2D points be given by:

$$
q_1 = \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}, \quad q_2 = \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix}, \quad q_3 = \begin{pmatrix} u_3 \\ v_3 \\ 1 \end{pmatrix}.
$$

Let the 3 3D points be given by:

$$
Q^m_1, Q^m_2, Q^m_3.
$$
Input and Unknowns

Given $q_i, Q_i^m, i = \{1, 2, 3\}$, and $K$ in the following equation:

$$q_i \sim KR \begin{pmatrix} 1 & -t \end{pmatrix} Q_i^m, i = \{1, 2, 3\}$$

Our goal is to compute the rotation matrix $R$ and the translation $t$. 
Goal: \((R, T)\)?
Given the three 3D points $Q_i^m$, $i = \{1, 2, 3\}$ we compute the 3 pairwise distances $d_{12}$, $d_{23}$, and $d_{31}$ as follows:

$$d_{ij} = \text{dist}(Q_i^m, Q_j^m)$$

$$\text{dist}(Q_i^m, Q_j^m) = \sqrt{(X_i^m - X_j^m)^2 + (Y_i^m - Y_j^m)^2 + (Z_i^m - Z_j^m)^2}$$
World frame to Camera frame

- Let the three 3D points $Q_i^m, i = \{1, 2, 3\}$ be denoted by $Q_i^c, i = \{1, 2, 3\}$ in the camera coordinate system.
- In other words, we have $Q_i^c = RQ_i^m - Rt$.
- Here $Q_i^m$’s are known variables and $Q_i^c$’s are unknowns.
- It is easy to observe the following since the distance between two points do not change when we transform them from one coordinate frame to another:

$$\text{dist}(Q_i^m, Q_j^m) = \text{dist}(Q_i^c, Q_j^c)$$
\[ \text{dist} (\mathbf{q}_i^m, \mathbf{q}_j^m) \rightarrow \text{gives} \]

the Euclidean distance between the two 3D points \( \mathbf{q}_i^m \) and \( \mathbf{q}_j^m \).
Reformulation of Pose Estimation

We can compute $Q_i^c$ as follows:

$$Q_i^c \sim K^{-1}q_i$$

$$Q_i^c = \lambda_i K^{-1}q_i$$

Here $\lambda_i$ is an unknown scalar that determines the distance of the 3D point $Q_i^c$ from the optical center along the ray $OQ_i^c$. 
Reformulation of Pose Estimation

We simplify the notations, let us denote $K^{-1}q_i$ as follows:

$$Q_i^c = \lambda_i K^{-1}q_i$$

$$K^{-1}q_i = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}$$ (1)
Reformulation of Pose Estimation

The pose estimation can be seen as the computation of the unknown $\lambda_i$ parameters.
Reformulation of Pose Estimation

\[ \text{dist}(Q_i^c, Q_j^c) = \text{dist}(Q_i^m, Q_j^m) = d_{ij}, \forall i, j = \{1, 2, 3\}, i \neq j \]

Euclidean distance between \( Q_i^c \) and \( Q_j^c \)

\[ \sqrt{(\lambda_i X_i - \lambda_j X_j)^2 + (\lambda_i Y_i - \lambda_j Y_j)^2 + (\lambda_i Z_i - \lambda_j Z_j)^2} = d_{ij} \]
Reformulation of Pose Estimation

We have 3 quadratic equations and 3 unknowns.

\[
\begin{align*}
(\lambda_1 X_1 - \lambda_2 X_2)^2 + (\lambda_1 Y_1 - \lambda_2 Y_2)^2 + (\lambda_1 Z_1 - \lambda_2 Z_2)^2 &= d_{12}^2 \\
(\lambda_2 X_2 - \lambda_3 X_3)^2 + (\lambda_2 Y_3 - \lambda_3 Y_3)^2 + (\lambda_2 Z_2 - \lambda_3 Z_3)^2 &= d_{23}^2 \\
(\lambda_3 X_3 - \lambda_1 X_1)^2 + (\lambda_3 Y_3 - \lambda_1 Y_1)^2 + (\lambda_3 Z_3 - \lambda_1 Z_1)^2 &= d_{31}^2 
\end{align*}
\]
Reformulation of Pose Estimation

\[
\begin{align*}
(\lambda_1 X_1 - \lambda_2 X_2)^2 + (\lambda_1 Y_1 - \lambda_2 Y_2)^2 + (\lambda_1 Z_1 - \lambda_2 Z_2)^2 &= d^2_{12} \\
(\lambda_2 X_2 - \lambda_3 X_3)^2 + (\lambda_2 Y_3 - \lambda_3 Y_3)^2 + (\lambda_2 Z_2 - \lambda_3 Z_3)^2 &= d^2_{23} \\
(\lambda_3 X_3 - \lambda_1 X_1)^2 + (\lambda_3 Y_3 - \lambda_1 Y_1)^2 + (\lambda_3 Z_3 - \lambda_1 Z_1)^2 &= d^2_{31}
\end{align*}
\]

- We have 3 quadratic equations and 3 unknowns.
- We can have a total of \(2^3\) possible solutions for the three parameters \((\lambda_1, \lambda_2, \lambda_3)\).
- Several numerical methods exist to solve the polynomial system of equations.
How to identify a unique solution?

- Out of the 8 solutions, only one will be the correct solution.
- In some of the solutions, the 3D point will be behind the camera.
- Using additional point correspondence, we can identify the correct solution.
Computing the Pose

- We remind you the relation between $Q^c_i$ and $Q^m_i$: $Q^c_i = RQ^m_i - Rt$.
- We are given $Q^m_i$ and we have computed $Q^c_i$.
- From three 3D-to-3D point correspondences we can compute the transformation parameters $(R, t)$ using Horn’s method.
Horn's method

Given two sets of 3D points in 2 different coordinate frames A and B, Horn’s method can be used to find a transformation $(R, T)$ between them $P^B_i = R P^A_i + T$

Frames: $A, B$

Indices: $i = \{1, 2, \ldots, 5\}$

Finals $(R, T)$
Reprojection error

Given 3D point \( Q^m_i, i = 1, \ldots, n \) and corresponding 2D points \( q_i, i = 1, 2, \ldots, n \), the reprojection error is given below.

Let the projection of \( Q^m_i \) on the image be given by

\[
\begin{pmatrix} u \\ v \\ w \end{pmatrix} \leftarrow KR(I - t)Q^m_i
\]

\[
q'_{i} = \begin{pmatrix} u/w \\ v/w \\ 1 \end{pmatrix}
\]

after removing the scaling factor.

Mean RPE

\[
\frac{1}{n} \sum_{i=1}^{n} \sqrt{(q_{ix} - q'_{ix})^2 + (q_{iy} - q'_{iy})^2}
\]
Presentation Outline

1. Review
2. Pose Estimation
3. RANSAC
Matching Images

Key points — interesting locations that are good for matching

Descriptors — vectors that describe a location

\[(x_i, y_i)\]

We match keypoints from left and right images.

To find the distance between two vectors — descriptors
We match keypoints from left and right images.

- One of the matches is incorrect!
- In a general image matching problem, we can have 100’s of incorrect matches.
Outliers and Inliers

We match keypoints from left and right images.
We match keypoints from left and right images.
Let us consider a simpler linear regression problem.

Problem: Fit a line to these datapoints

Least squares fit

How can we fix this?

Slide: Noah Snavely
Idea

- Given a hypothesized line.
- Count the number of points that agree with the line, i.e., points within a small distance of the line.
- For all possible lines, select the one with the largest number of inliers.

Slide: Noah Snavely
Counting Inliers

Slide: Noah Snavely
Counting Inliers

- 3 inliers

Slide: Noah Snavely
20 inliers!

Slide: Noah Snavely
How do we find the best line?

- Unlike least-squares, no simple closed-form solution
- Hypothesize-and-test
  - Try out many lines, keep the best one
  - Which lines?
RANSAC for Pose Estimation

Given:
- 2D points $q_i$, $i = 1, \ldots, n$
- 3D points $q_i^m$, $i = 1, \ldots, n$

Some of the matches are incorrect. Use RANSAC to find incorrect matches.

$$\text{POINT-SET} = \{1, 2, \ldots, n\}$$

$$\text{TRIPLET} \leftarrow \text{RANDOM SET OF 3 INDICES } (i, j, k)$$

$$(R, T) \leftarrow \text{POSE} (q_i, q_j, q_k, q_i^m, q_j^m, q_k^m)$$

You get 8 solutions for every triplet. Use RPR (reprojection error) to select the inliers (correct correspondences).
RANdom SAmple Consensus

Slide: Noah Snavely
RANdom SAmple Consensus

\[ T = \begin{pmatrix} x'_{i1} \\ y'_{i1} \end{pmatrix} - \begin{pmatrix} x_{i1} \\ y_{i1} \end{pmatrix} \]

Slide: Noah Snavely
RANdom SAmple Consensus

Select one match at random, count inliers

Slide: Noah Snavely
RANdom SAmple Consensus

Select one match at random, count inliers

Slide: Noah Snavely
RANdom SAmple Consensus

Select one match at random, count inliers

Slide: Noah Snavely
RANdom SAmple Consensus

Select one match at random, count inliers

Slide: Noah Snavely
RANdom SAample Consensus

Select another match at random, count *inliers*

Slide: Noah Snavely
RANdom SAmple Consensus

Select another match at random, count *inliers*

Slide: Noah Snavely
Select another match at random, count inliers

Slide: Noah Snavely
RANdom SAmple Consensus

Output the translation with the highest number of inliers

Slide: Noah Snavely
Idea:

- All the inliers will agree with each other on the translation vector; the (hopefully small) number of outliers will (hopefully) disagree with each other
  - RANSAC only has guarantees if there are $\leq 50\%$ outliers
- All good matches are alike; every bad match is bad in its own way - Alyosha Efros, CMU

Slide: Noah Snavely
The 4 corner points for each window is matched with 3D points.

Largest

Inliers? \{1, 2, 3\} \sim \{4, 5, 6\}
Camera Pose Estimation and RANSAC
Srikumar Ramalingam

Pose Estimation

Inliers?
\( \{2, 3\} \)

Largest

\( \text{Inliers?} \)
Pose Estimation

- Inliers? \{1, 2, 3\}

Largest
Pose Estimation

Inliers?

Largest

\[ \{4, 5, 6\} \]
RANSAC

- **Inlier threshold** related to the amount of noise we expect in inliers
  - Often model noise as Gaussian with some standard deviation (e.g., 3 pixels)
- **Number of rounds** related to the percentage of outliers we expect, and the probability of success we would like to guarantee
  - Suppose there are 20% outliers, and we want to find the correct answer with 99% probability
  - How many rounds do we need?
How do we generate a hypothesis?

Slide: Noah Snavely
General Version - RANSAC

1. Randomly choose \( s \) samples
   - Typically \( s = \) minimum sample size that lets you fit a model
2. Fit a model (e.g., line) to those samples
3. Count the number of inliers that approximately fit the model
4. Repeat \( N \) times
5. Choose the model that has the largest set of inliers

Slide: Noah Snavely
How many rounds?

<table>
<thead>
<tr>
<th>s</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
<th>25%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>19</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>5</td>
<td>9</td>
<td>13</td>
<td>17</td>
<td>34</td>
<td>72</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>6</td>
<td>12</td>
<td>17</td>
<td>26</td>
<td>57</td>
<td>146</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>7</td>
<td>16</td>
<td>24</td>
<td>37</td>
<td>97</td>
<td>293</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>8</td>
<td>20</td>
<td>33</td>
<td>54</td>
<td>163</td>
<td>588</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>9</td>
<td>26</td>
<td>44</td>
<td>78</td>
<td>272</td>
<td>1177</td>
</tr>
</tbody>
</table>

$p = 0.99$

- If we have to choose $s$ samples each time
  - with an outlier ratio $e$
  - and we want the right answer with probability $p$

Slide: M. Pollefeys
Acknowledgments

Some presentation slides are adapted from the following materials:

- Peter Sturm, Some lecture notes on geometric computer vision (available online).
- Kristen Grauman’s computer vision lecture slides
- Noah Snavely’s computer vision lecture slides